

In the Specification

Please refer to Appendix A for mark-ups reflecting changes to the specification.

On ~~Page~~ 1 after the title, please insert the following:

RELATED APPLICATIONS

A¹ This application claims priority under 35 U.S.C. § 119 to U.S. Provisional Application No. 60/130,735 filed April 20, 1999.

On ~~page~~ 12, please replace the paragraph beginning at ~~line~~ 8 with:

A² The computer moderated electronic commerce system of the present invention also allows simultaneous multithread operation, as is depicted in block diagram in pictorial form in **Figure 4**. As is shown, a data server 260 may be utilized to allow database connection pooling. This optimizes the efficiency of database resource utilization. In the preferred embodiment, an ORACLE database may be utilized. The data server 260 provides the data to the application server layer which is composed of a number of separate components, some of which were discussed above in connection with **Figure 2**. As is shown, sales data server 204 is communicatively and operationally coupled with sales business object server 202 which is capable of supporting a number of simultaneous electronic commerce communication sessions including sessions 270, 272, 274, 276. The sales business object server 202 is communicatively and operationally coupled to sales Internet server 280 which is in turn communicatively and operationally coupled to web server 282. Web server 282 allows for the simultaneous communication with potential buyers at and utilizing computing devices 290, 292, 294, 296, and 298. Essentially, sales business object server 202 is a multithread server which utilizes multiprocessor servers. Multiple user sessions may be enabled in one process, which makes for efficient use of the operating system resources. The utilization of web server 282 and Internet sales server 280 facilitates relatively low-cost simultaneous communication with buyers and potential buyers. The communication between web server 282 and the computers (290, 292, 294, 296, and 298) which are under control of the buyers and potential buyers is conducted utilizing a relatively low bandwidth HTML dialog.

On ~~page 13~~, please replace the paragraph beginning at ~~line 22~~ with:

A³

FIGURE 5 is a block diagram and pictorial representation of an overview of the interface options available utilizing the present invention which represents still another commercially advantageous product feature. The view of **FIGURE 5** represents the different layers of processing which must be implemented in order to provide the system of the present invention. As shown, the data server 260 maintains the metadata which is utilized during the commercial transaction. This may include component identification and component property of the various items offered for sale by the seller. The data server 260 communicates with the sales data server 204. The sales data server 204 communicates with the sales business object server 202. The sales business object server 202 may communicate over communication link 302 with data processing system 296. The communication between the sales business object server 202 and data processing system 296 may be conducted utilizing the JAVA programming language; therefore, the client-user interface is supported by JAVA and communication is conducted over a local area network or a high speed wide area network.

On ~~page 14~~, please replace the paragraphs beginning at ~~lines 11 and 21~~, respectively, with:

A⁴

An alternative means of communicating with buyers is also provided. As is shown in **Figure 5**, sales internet server 280 may also communicate with sales business object server 202. Web server 282 in turn communicates with sales internet server 280. Web server 282 communicates intermittently over communication link 300 utilizing an HTTP protocol. Communication is conducted with one or more data processing systems such as data processing system 290. Data processing system 290 is operating in a Web browser mode of operation. Communication between data processing system 290 and Web server 282 is conducted utilizing HTML. The communication link 300 is a low-speed wide-area network or internet connection. The view of **Figure 5** shows two different user interfaces which are supported by the data maintained in data server 260. The dialog which makes up the content of the electronic transaction is constructed dynamically by the sales data server 204 and sales business object server 202 utilizing the metadata maintained on data server 260. In this way, a single data base may be utilized to support a plurality of simultaneous transaction dialogs in a plurality of differing programming interfaces supported by different programming languages and over communication links with different capabilities.

On page 16, please replace the paragraph beginning at line 2 with:

A⁵

FIGURE 10 is a pictorial and graphic representation of the maintenance of the various user interfaces in accordance with the preferred embodiment of the present invention. As is shown, a product rule set 310 is utilized to map or organize the various product features and options which are offered by the seller to potential buyers. The Dialog Controls module 312 is utilized to control the content of the dialog boxes of a graphical user interface. The content may be edited utilizing Dialog Editor 314. In accordance with the preferred embodiment of the present invention database 260 maintains the metadata associated with the product line and options which are set forth in the product rule set 310. Database 260 supports a variety of rendering engines which are utilized to generate graphical user interface screens as part of the dialog transaction. As is shown, JAVA-rendering engine 320 accesses database 260 in order to generate dynamically and in real time user interface 326 which is a JAVA applet. HTML-rendering engine 322 utilizes database 260 to generate dynamically and in real time graphical user interface screen 328 which is an HTML user interface. C++-rendering engine 324 utilizes data from database 260 to generate dynamically and in real time graphical user interface 330. Note that graphical user interfaces 326, 328, and 330 are all very similar in their content and layout, even though they are generated in different operating environments utilizing different programming languages. All of the rendering engines 320, 322, and 324 make dynamic use of the Dialog Control box 312 in order to generate the graphical user interfaces 326, 328 and 330.

On page 17, please replace the paragraph beginning at line 12 with:

A⁶

FIGURE 11 is a flowchart representation of dynamically construct a series of graphical user interfaces, in real time, and utilizing a single database. The process starts at block 404, wherein communication is established between a buyer and a seller. This communication will occur over a particular communication channel. For example, the buyer may make contact with the seller utilizing the Internet by accessing the seller's website. Alternatively, the buyer may dial-in to a wide area network utilizing conventional telecommunications modem connections in order to communicate with the seller utilizing a graphical user interface constructed in the conventional manner utilizing the C++ programming language. Alternatively, the potential buyer may make communication with the seller utilizing JAVA applets. All three of these scenarios are alternatives to one another and are graphically depicted in **FIGURE 10**. In accordance with step 406, the seller's data processing system

Ak
activates the appropriate rendering engine which is suited for the mode and channel of communication which has been established by the buyer. For example, the buyer is making contact utilizing the Internet, the HTML rendering engine will be activated. Then, in accordance with step 408, the seller's data processing system calls the appropriate dialog manager module. In actual practice, an electronic transaction is composed of a series of cascading and logically-linked graphical user interfaces. Each graphical user interface has associated with it a particular dialog manager module, as is conventional. The dialog manager module identifies each and every component of the graphical user interface which will be presented to the buyer. Some of the components are "inactive" components and merely present images, data, or information; however, other elements of the graphical user interface are "active" elements which are adapted to receive user input typically through the detection of the operator actuation of the graphical pointing device (typically the depression of the left button on the mouse associated with the buyer's computer). Then, in accordance with step 410, the seller's data processing system utilizes the rendering engine to generate an associated graphical user interface. Concurrently with this step, and as is set forth in step 412, the data processing system of the seller communicates with a single database in order to read metadata which is associated with the graphical user interface. As described above, the metadata may be arranged utilizing conventional tools such as a rule maintenance module. The metadata may comprise 17 simple product number and feature information; however, in alternative electronic transactions, the metadata may comprise a substantial body of transaction and product information. The more complex the subject matter of the electronic transactions, the more likely there is to be associated with each graphical user interface a greater amount of detailed information. This communication between the rendering engine and the single database is conducted in real time and is done so dynamically during the interaction between the seller's data processing system and the buyer's data processing system. One significant advantage of this approach is that the seller need not maintain multiple parallel databases for each rendering engine; instead, a single database may be maintained. This is a low cost option since there are greater costs associated with maintaining several parallel databases and it is often difficult to maintain consistency between such databases.

On page 19, please replace the paragraph beginning at line 4 with:

At
In accordance with step 414, the data processing system under the control of the seller applies the metadata dynamically and in real time during creation of the graphical user interface. Then, in accordance with block 416, the seller's data processing system monitors for the dialog in order to determine input or selection of options by the buyer. In accordance with step

A⁷

418, if necessary, the data processing system under the control of the seller is utilized to write data to the database. Then, in accordance with block 420, the data processing system which is under the control of the seller monitors for a termination of the communication session. If the session is ended, the process ends. However, if the session is not ended, the data processing system under control of the seller monitors the dialog as conducted through the dialog boxes and returns to block 408, wherein the appropriate dialog monitor module is called based upon the buyer's input. The process repeats over and over again until the dialog is terminated. As a consequence of the dialog, a substantial amount of detailed metadata is sequentially presented to the potential buyer in a series of cascading graphical user interface dialog boxes. In turn, the data and selections provided by the buyer may be recorded to the single database in order to enable completion of the transaction or return to the transaction at a later date. It is widely known that many electronic interactions do not result in a transaction at the first contact. It may take several interactions with the buyer before a transaction is completed. This is the reason that most electronic transaction systems have shopping carts which may be preserved in memory and recalled at a later date by the buyer in order to allow the buyer to modify or add to the shopping cart.

On ~~page 25~~, please replace the paragraph beginning at ~~line 1~~ with:

A⁸

The dialog controls module 514 is comparable to a dialog controls module present in the "rhythm" product sold by i2 Technologies, Inc. Assignee of the present application. This module allows one to connect structure boxes to dialog controls one has created utilizing the dialog editor 516. An exemplary screen from the dialog controls module 514 of **FIGURE 12** is depicted in **FIGURE 7**. As is shown, a "NAME 601" field is provided which displays the name of the active structure box. Additionally, a "CODE FIELD 603" is provided which displays the code of a selected box. One can select another code from a drop down list. The boxes are listed in the same order as they appear in the structure. When a particular code is identified in Code Box 603, a grid 605 is displayed there below which displays all dialogs, controls and variables that have been earlier designated as being associated with the box or boxes. As shown in **FIGURE 7**, a grid 605 includes a variable field 607, a dialog ID field 609 and a control field 611.

On ~~page 26~~, please replace the paragraph beginning at ~~line 21~~ with:

A⁹

Figure 13 is a table which identifies the primary values which are under the parameters control. A "CLASS" value cannot be edited. However, the X and Y coordinates may be edited. These X and Y coordinates establish

A⁹

an X and Y coordinate in pixels for a graphical user interface element. The Y variable displays the width of the control element in pixels and the value H displays the height of the control element in pixels. The "TITLE" element is a field for displaying the title of the control. The "OPTION" group cannot be edited. The "TEXT" element identifies a default static control text. The "FONT" element establishes a font name, size, and style. The "MULTI-LINE" element identifies an edit control feature in which a multiple or single line text may be identified. The "RTF" and "RTF TOOLBAR" elements identify a text box that can contain RTF text (MICROSOFT'S rich text format). The "EXTENDED" element identifies a multi-select list box that includes a check box for items in that list box.

On ~~page 27~~, please replace the paragraph beginning at line ~~21~~²³ with:

A¹⁰

Figure 14 is a pictorial representation of how elements in a graphical user interface in an HTML environment are laid-out in accordance with the preferred embodiment of the present invention. As is shown, HTML page 700 is divided up into segments which cover the entire space that is available. In the HTML format, this is considered to be a irregular table. The space within HTML page 700 will be accounted by a "cell" within the table even if there is no text or image provided in that particular portion of Page 700. In the view of **Figure 14**, "cells" which do not contain any text or images are shown in dashed outlined form. Portions of the graphical interface which include text or images are shown in solid line form. The space within HTML page 700 may be divided up into rows. The first row is made up of cells 701, 703, 705, 707. Of these, cell 707 does not include any text or image portions. It is merely space which is not utilized. The cell 707 is utilized by the program in order to "account" for the space. The second row includes cells 709, 711, and the upper portion of cell 713. Cell 711 does not contain any text or images and is utilized merely to account for the space between cell 709 and cell 713. The third row is made up of cells 715, 717, 719, 721, and 713. Of these, cells 715, 721 are regions which do not contain any text or images and which are utilized only to account for the space in that row. The next row includes cells 723, 725, 727, 729, 731. Cells 725, 729 include text and/or images while cells 723, 727, 731 do not include any text or images and are merely utilized to account for the space in that row. This row also includes the portion of cell 713. The next row is made of cells 733, 735, 737, 739, 741, 713. Of these cells, cells 733, 737, 741 are not utilized for depicting any text or images and are merely utilized to account for the space in that row. The next row includes cells 743, 745, 747, 749, 751 and 713. Cells 745, 749, 713 include text and/or images. Cells 743, 747, 751 do not include text or images. The next row includes cells 753, 755,

A¹⁰

757. Cells 755, 757 include text and/or images while cell 753 does not include any text or images. The final row, is made of cell 759.

On ~~page 29~~, please replace the paragraph beginning at ~~line 7~~ with:

A¹¹

A routine which implements the preferred embodiment of the present invention is depicted in the flow chart of **Figure 16**. The process starts at Block 801, and goes to Block 803, wherein the product model is loaded. Next, in accordance with Block 805, the Dialog Controls Module is loaded. Then, in accordance with Block 807, the Dialog Manager is loaded. Next, in accordance with Block 809, the Dialog Controls Module is searched for the particular product model. Once it is located, the values associated with that particular product model are read from the Dialog Controls Module. Then, in accordance with Block 813, the size of the user interface elements are adjusted if necessary in order to have the "cells" sized to an extent sufficient to carry all of the text and/or images which are to be posted in that particular portion of the HTML page. This can be best understood with reference again to **Figure 14**. For the third row from the top, the text resident in cells 717, 719 may require a greater or lesser amount of space. If these cells need to be expanded, space that is not dedicated for some other purpose can be utilized. For example, cells 717, 721 may be further reduced in order to allow a greater amount of text or images to be posted to that portion of the HTML page 700. Sizing issues may become more complex if fonts sizes are changed for a particular graphical user interface. Changes in font size will necessitate recalculation of the available cell space and enlargement of the cells dedicated for text in order to accommodate the enlarged text.

On ~~page 30~~, please replace the paragraph beginning at ~~line 1~~ with:

A¹²

Returning now to **Figure 16**, once the user interface elements are adjusted in size, then the user interface is populated with its elements in accordance with Block 815 and the process ends at Block 817.
